Highly Dependable and Affordable Computing Systems for Long-Life Deep-Space Applications

Leon Alkalai
Jet Propulsion Laboratory
Pasadena, CA 91109

Ann T. Tai
IA Tech, Inc.
Los Angeles, CA 90024

Faster, Better, Cheaper: NASA's New Standard

With NASA's spectacular return to Mars on July 4th 1997, the Mars Pathfinder Lander and its Sojourner Microrover have set a new standard for *Faster, Better, Cheaper* space exploration missions. (To spell out a little bit the new standard here?) X2000, a new-generation space technology program aimed at providing an engineering model to multiple long-life deep-space missions [1], is intended to realize the new standard by achieving at least an order of magnitude improvement in both performance and dependability under stringent power, weight and cost constraints. Currently, five missions including Pluto/Kuipar Express, Europa Orbiter, Mars Sample Return, Champollion/DS4, and Solar Probe have determined to adopt the X2000 technology. Thus X2000 is anticipated to supply benchmarks for NASA's new standard.

High-Assurance Criteria: A Spaceborne System Perspective

Due to the multi-mission objective, the architecture of the X2000 spaceborne computing system is driven by a comprehensive set of criteria. The key criteria include

Reliability, availability Reliability refers to a system's ability to continuously serve a long-life mission (in general, 5 to 10 years); availability means the readiness to serve the mission in a deep-space unsurveyed environment (e.g., the planet Pluto).

Evolvability, maintainability Evolvability is defined as a spaceborne computing system's on-board capability to accomplish enhancement of dependability, performance and

Sp

functionality with minimal ground support. Maintenability is an evolvable system's ability to preserve its performance and fault tolerance properties, and to adjust those properties to mission environment variation.

Affordability refers to the cost-effectiveness of a system for which performance, fault tolerance and other operational attributes are optimized under stringent low-cost, low-power, low-mass and low-volume criteria.

Miniaturization is the design criterion for a highly integrated microspacecraft system that includes all the science and engineering functions into a single modular and scalable architecture ("spacecraft on a chip" [1]).

Note that evolvability and maintainability are fresh notions introduced by the new-generation of spaceborne system. Associated issues include *reconfigurable hardware* and *upgradable software.* "Reconfigurable hardware" refers to the on-board ability to create, under the guidance of artificial intelligence (such as genetic algorithms for FPGA self-reconfiguration), new electronic functionalities required by the conditions that are unanticipated but appear during a mission's long-life. "Upgradable software" aims for allowing flight software to be developed and improved incrementally during a mission's long life span, such that 1) time to first version completion and installation can be significantly reduced, and 2) uploading a particular software module can be conducted without the costly interruption of mission's normal functions (the current approach is to replace the entire flight software, which would prevent the system from functioning for an appreciable amount of time, typically several hours). Moreover, upgradable software permits a spaceborne system, during its mission's long life span, to keep pace with the latest software technologies for better performance, fault tolerance and functionality, instead of being constrained by those available prior to mission launch. With an evolvable system, on-board maintenance that assures the consistency between old and new versions with respect to their functionalities, performance attributes and fault tolerance mechanisms is crucial. NASA experienced a gap in fault tolerance protection on April 10, 1981, when a timely synchronization check was omitted after the addition of an alternate reentry program [2]. As a result, the first flight of the US space shuttle program was aborted 19 minutes before launch.

Striving for an Integrated Approach

The above brief discussion indicates that various interactions exist among the high-assurance criteria of spaceborne systems. On one hand, these criteria are mutually conducive. Examples in this regard are 1) evolvability and miniaturization reinforce affordability, and 2) maintainability reinforces reliability and availability. On the other hand, these criteria interfere each other. Examples of the interference include i) affordability imposes constraints on the means for achieving reliability and availability, and ii) the availability criterion makes on-board maintenance a more difficult process. Therefore, in our view, high-assurance system engineering practice shall 1) enhance the effects from the mutual reinforcement among various criteria, and 2) eliminate or minimize the effects from their mutual interference. In other words, the various interactions call for an *integrated approach*, meaning that the methods for accomplishing multiple criteria shall be coordinating instead of interfering. With an integrated approach, it is necessary to analyze the relationships among the criteria, including 1) to qualitatively predict the types of interactions among them, and 2) to quantitatively evaluate the collective benefits and risks from the techniques for assuring each individual goal. Accordingly, it is important to further investigate the methods that facilitate coordinations among high-assurance techniques, including integrated measures such as performability [3, 4].

Challenges in Evolvability

Among other research opportunities the new generation of spaceborne system exhibits to us, a particular challenging subject is to assure reliable spaceborne system evolution, namely, reliable hardware reconfiguration and software upgrading. The means for the assurance are two-fold: i) to thoroughly verify and validate the initial system, its genetic algorithms and highly modularized software components implemented for upgrading, and ii) to eliminate or minimize the adverse effects from reconfiguration or upgrading due to the residual faults in a genetic algorithm or an upgraded software version. The latter is referred to as "on-board maintenance" which encompasses 1) *preventive maintenance* — removing and minimizing error conditions before they produce symptoms, and 2) *corrective maintenance* — detecting and correcting the error conditions due to residual fault manifestation. In the context of long-life deep-space missions, the means for preventive maintenance include periodic rejuvenation via system reset and software re-initialization, while corrective maintenance between upgrading activities can be realized via utilizing appropriate fault tolerance techniques. In turn, on-board maintenance should also be a multiple criteria driven process. That is, the means for assuring reliable evolution are not permitted to sacrifice affordability, or to violate power and mass constraints. In our view, affordable on-board maintenance can be achieved by exploiting inherent, non-dedicated resource redundancies on board in order to simultaneously meet the low power, low weight, low cost and high reliability criteria. Examples of such redundancies include 1) processing and storage elements in a parallel/distributed system architecture, and 2) multiple software versions uploaded to a spaceborne system through upgrading activities. A potential utility of inherent software redundancy is the following: To assure reliable software upgrading on-board, an earlier and relatively more mature version can be utilized as a backup module, upon the detection of an error caused by a residual software defect in the updated version, the backup module becomes responsible to restart the failed task based on checkpointing. In the case where an error is detected in a newly introduced function such that there does not exist a corresponding backup module, forward recovery techniques shall apply. Therefore, the mechanisms that support checkpointing activities specific for an evolvable distributed computing environment, such as task allocation, load sharing, rollback and roll-forward recovery, require innovative solutions for the assurance of reliability gain.

References

[1] L. Alkalai, "NASA Center for Integrated Space Microsystems," in *Proceedings of Advanced Deep Space System Development Program Workshop on Advanced Spacecraft Technologies*, (Pasadena, CA), June 1997.

[2] A. Avižienis, "Towards systematic design of fault-tolerant systems," *IEEE Computer*, vol. 30, pp. 51–58, Apr. 1997.

[3] J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Trans. Computers*, vol. C-29, pp. 720–731, Aug. 1980.

[4] A. T. Tai, J. F. Meyer, and A. Avižienis, *Software Performability: From Concepts to Applications.* Boston, MA: Kluwer Academic Publishers, 1996.